

Utilizando nosso tipo monádico

Transcrição

Vamos remover os testes que fizemos em `app/app.js` para em seguida alterarmos nossa aplicação para fazer uso da *maybe monad*.

Vamos importar `Maybe` em `app/nota/service.js`, alterar o método `list()` para retornar uma mônada encapsulando o tipo `null`. Por fim, vamos alterar todas funções usadas na composição de `sumItems` para que recebam o tipo monádico:

```
import { handleStatus } from '../utils/promise-helpers.js';
import { partialize, pipe } from '../utils/operators.js';
// importou Maybe
import { Maybe } from '../utils/maybe.js';

const API = 'http://localhost:3000/notas';

// adequou cada função para receber o tipo monádico

const getItemsFromNotas = notasM => notasM.map(notas => notas.$flatMap(nota => nota.itens));

const filterItemsByCode = (code, itemsM) => itemsM.map(items => items.filter(item => item.codigo ==

const sumItemsValue = itemsM => itemsM.map(items => items.reduce((total, item) => total + item.valor

export const notasService = {

  listAll() {

    return fetch(API)
      .then(handleStatus)
      // retorna um `Maybe`
      .then(notas => Maybe.of(null))
      .catch(err => {
        console.log(err);
        return Promise.reject('Não foi possível obter as notas fiscais');
      });
  },

  sumItems(code) {
    const filterItems = partialize(filterItemsByCode, code);

    const sumItems = pipe(
      getItemsFromNotas,
      filterItems,
      sumItemsValue
    );

    return this.listAll()
      .then(sumItems)
      // obtendo o valor da mônada
      .then(result => result.getOrElse(0))
  }
};
```

```
    }  
  };  
}
```

Excelente. Agora, o resultado será `0`, sem quebrar nossa aplicação. Depois do teste feito, altere o método `list()` passando o valor retornado pela API para a `mônada`:

```
listAll() {  
  
  return fetch(API)  
    .then(handleStatus)  
    .then(notas => Maybe.of(notas))  
    .catch(err => {  
      console.log(err);  
      return Promise.reject('Não foi possível obter as notas fiscais');  
    });  
},
```